control of access rights according to the present invention will be described with reference to the accompanying drawings.

The FIG. **1** gives an overview of the method for controlling access rights. A set of subjects **1** as holders of access rights is defined and associated to a set of role types **2**. The role types **2** are instantiated into a set of role instances **3** and therefore the subjects **1** are associated to the role instances **3**. Multiple subjects **1** can be associated with one role type **2**. Also, a subject **1** can be associated with more than one role type **2**. The instantiation of role types **2** into role instances **3** also determines the association between the role instances **3** and the objects **4** of the computer system. Usually there will be multiple instances of one role type due to different parameter values provided by different subjects.

The FIG. **2**A gives an overview for the method of role type instantiation. Persons **5** that are users of an enterprise computer system are employees acting in assigned job positions **6**. Each job position **6** is associated with a set of functional tasks and, thus, these tasks are associated with users in the enterprise organization hierarchy. Each task requires a set of competencies, which can be viewed as a set of specific access rights to a set of objects **4** necessary to carry out that task. Hence, each job position **6** ultimately associates a user with specific access rights to a set of objects **4**. Thus, a security administrator must be able to associate these rights, objects, and transactions with the job positions of the enterprise organization. To enable this, the concepts of role types and role instances are defined.

The FIG. **2**B shows job positions **6**, role types **2**, and the creation of role instances **3**. The diagram shows an organization structure, e.g. organization units **7** and job positions **6**, on the left and a set of role types **2** on the top of the matrix. An "X" in a field of the matrix means that a role instance **3** of the corresponding role type **2** is assigned to the job position **6**. The necessary parameter values to instantiate the role type **2** are derived from attributes of the individual job position **6** or a higher level organization unit. The values of these attributes determine the actual competencies the job position **6** is assigned via the role instance **3**. Job positions **6** may share the same role instance **3** as illustrated by the shaded fields in a column.

A job position **6** is associated with one or more role instances **3**, depending upon how granular the job position **6** is intended to be. These role instances **3** are derived from different role types **2**. For example, there are three role instances associated with the job position "staff member **2**" of "private loans", one derived from the role type "loan specialist", another one derived from "customer consultant", and one derived from "bank employee".

Often similar job positions, such as "staff member **1**" and "staff member **2**" of the "private loans" department, will be assigned to the same role instance as shown from the shaded fields in the matrix, because none of the attributes that are relevant for instantiating the role type differ between the job positions. However, different job positions **6** or similar job positions **6** in different organization units **7** will usually be associated with different role instances **3** of the same role type **2**, because they bring in different attribute values for the role type instantiation. In the above example the role type "loan specialist" is instantiated in two different role instances that are bound to two different job positions of the department "object appraisal", the "team-leader" and the "staff member **1**" position.

Job sharing can be modelled by assigning one job position **6** to multiple persons **5**. On the other hand a single person **5** may be assigned to multiple job positions **6**. For example, a person **5** in a "staff member" position in a department may also act, perhaps temporarily, as the "department manager". Of course, assignment to some job positions **6** may exclude assignment to other job positions **6** for separation-of-duty reasons. For example, a person **5** in the job position **6** "security administrator" may not be assigned to the job position **6** of "auditor" because otherwise the accountability of the "security administrator's" actions would be lost.

The FIG. **2**C shows an example of the role type instantiation method in more detail, especially for the role instance in the framed matrix cell **15** of FIG. **2**B. A role instance **3** binds the relative competencies defined by a role type **2** to the objects **4**, and access rights specific to an organization unit **7** or a job position **6**. To perform this, at first for each organization unit and for each job position **6** a set of attributes has to be declared as relevant for role type instantiation. These attributes are said to be advertised. As an example, this could be the department identity or the location attribute of the department organization unit or the project identity attribute of a job position **6**. Second, so-called relative resource sets **8** may be defined and associated with role types **2**. A relative resource set **8** specifies the parameters it expects for instantiation from among the advertised ones in the enterprise. For example, one could define the relative resource set "printers" (printlocation) by enumerating the printers that are available to each location:

printers (Boeblingen): ={p**2160**, p**2240**, . . . }

printers (Heidelberg): ={prt**01**, prt**02**, . . . }

The "print location" parameter is declared as referencing the advertised "location" attribute of a department.

Thus, when a job position **6** as part of certain organization units **7** is combined with a role type **2** associated with parameterized relative resource sets **8**, the actual resources can be determined by instantiating the parameters with the values of the advertised attributes for this job position **6**. In the example of FIG. **2**C, if

1. private loans is located in Heidelberg,
2. the relative resource set **8** "printers (printlocation)" is associated with role type **2** "bank employee" with permission "use", and
3. "staff member **1**" of the department "private loans" is assigned the role type **2** "bank employee".

Then "staff member **1**" will have "use" access to the printers "prt**01**, prt**02**, . . . ".

Whether a new role instance **3** has to be created in this case depends on whether the "bank employee" role type **2** has already been instantiated with the same parameters. If this is the case "staff member **1**" will only be assigned the already existing role instance **3** "bank employee (. . . , Heidelberg, . . .)".

FIG. **3**A shows the role type hierarchy in the disclosed inventive method. The access-control policy semantics captured by the specification of role types reflect the functional partitioning and inclusion of generic access rights, resources, and transactions necessary to conduct the business activities and management of an enterprise. This partitioning and inclusion is intended to cover the data and application access relationships that are independent of the users job position **6** and organization context, i.e. units **7**, of the enterprise. The rest of the access-control semantics captured by role instances **3** and job positions **6** reflect constraints placed by enterprise policies, such as the need-to-know and separation-of-duty policies, on enterprise organization units **7**.

A role type **2** is defined as a set of generic parameter-dependent resources and their associated permissions or